

## 1. 如何阅读本章

LPC111x系列处理器Flash配置的差异如表 17-213所列。

表 213. LPC111x flash 配置

型号	Flash
LPC1111	8 kB
LPC1112	16 kB
LPC1113	24 kB
LPC1114	32 kB

**注释：**除了 ISP 和 IAP 命令，Flash存储器的访问时间可以在Flash控制块的一个寄存器中配置，具体请查阅 17-10小节。

## 2. 引导加载程序

引导加载程序（BootLoader）在系统复位后执行初始化操作，并且提供完成Flash存储器编程的方式。它可执行空处理器的初始化，也可对一个已写入程序的处理器进行擦除或重写，也可以在系统执行阶段对 Flash 存储器进行编程。

## 3. 特性

- 在线编程：在线编程 (ISP) 是使用引导加载程序和异步串口，对片上Flash进行编程或重新编程。可以通过这种方式向用户板烧写程序。
- 现场编程:现场编程 (IAP) 是指应用代码直接执行对片上Flash存储器的擦除和编程。
- Flash 访问时间可以通过Flash 控制模块一个寄存器来配置。

## 4. 应用

引导加载程序（BootLoader）提供在线编程（ISP）和现场编程（IAP）两种实现芯片Flash存储器编程的方式。

## 5. 描述

引导加载程序在每次系统上电或复位时执行。引导加载程序可以执行ISP 命令处理程序或用户应用代码。复位后如果在PIO0\_1引脚出现一个低电平，被当作开始ISP命令处理程序的外部硬件请求。当RESET引脚的上升沿信号到来时，电源引脚处于正常电平，需要对PIO0\_1信号采样3 ms以上，以确定是执行用户代码还是ISP处理程序。如果 PIO0\_1

采样为低, 且看门狗溢出标志被置位, 则开始 ISP 处理程序的外部硬件请求被忽略。如果没有 ISP 请求 (复位后PIO0\_1为高电平), 则开始寻找有效的用户程序。如果找到有效的用户程序, 则将执行权限转给该用户程序。如果没有找到有效的用户程序, 则调用自动波特率程序。

PIO0\_1引脚作为 ISP 硬件请求时要特别注意。由于 PIO0\_1 复位后处于高阻态, 需要用户提供外部硬件 (上拉电阻或其它器件) 使该引脚处于定义的状态。否则可能不能如预进入 ISP 模式。

## 5.1 复位后的存储器映射

系统引导块共占用16 kB。引导块占据从地址0x1FFF 0000开始的一段存储区域。系统引导程序设计为从此处开始运行, 但是 ISP和IAP程序使用部分片上 RAM。如何使用RAM在稍后的小节中会描述。复位后, 片上Flash引导块中的中断向量表也会被激活, 也就是说, 从地址0x0000 0000开始的512字节的存储器区域也是可见的。

### 5.1.1 有效用户代码的判定标准

有效用户代码的判定标准: 保留的 Cortex-M0 中断向量位置 7 (在中断向量表中偏移地址为0x 0000 001C) 应该包含向量表入口0到6的检验和的补码。这将导致前 8 个表入口的检验和为 0。引导程序代码检查Flash扇区0的前8个位置的检验和。如果结果为0, 则将执行权限交给用户代码

如果结果无效, 自动波特率程序通过串口 0 与主机进行同步。主机发送一个 '?' (0x3F) 作为同步字符然后等待应答信号。主机方串口设置应该是 8 位数据、1位停止位无校验位。自动波特率程序按自己的频率计算收到同步字符的位时间, 然后对串口波特率发生器进行编程。它也发送一个ASCII 字符串 ("Synchronized<CR><LF>") 到主机。作为应答主机也发送同样的字符串 ("Synchronized<CR><LF>")。自动波特率程序检查接收到的字符串以确认同步。如果同步以确认则发送 "OK<CR><LF>" 字符串到主机。主机将以自己正在运行的晶振频率作为应答。例如, 主机运行在 10 MHz, 主机应答应该是 "10000<CR><LF>"; 接收到晶振频率后, "OK<CR><LF>" 字符串被发送到主机。如果同步没有被确认, 自动波特率程序仍然等待同步字符。为确保自动波特率程序能够正确运行而不调用 ISP, CCLK 频率必须大于等于10 MHz。

一旦晶振频率接收部分完成, 将调用 ISP 命令处理。为了安全考虑, 在执行flash 擦写操作之前和执行 "Go" 命令之前, 应该调解锁命令 "Unlock"。其他命令不用调解锁命令。每次 ISP 会话只需要调用执行一次解锁命令。解锁命令的详细描述见 [第227页的 17-7 “ISP 命令”小节](#)。

## 5.2 通信协议

所有的 ISP 命令都必须以 ASCII 字符串的形式发送。字符串必须以回车 (CR) 和/或换行(LF) 控制符结尾。额外的 <CR> 和<LF> 控制符将被忽略。所有的 ISP 应答信号也以 <CR><LF> 控制符结尾的 ASCII 字符串形式发送。数据以 UU-encoded 格式发送和接收。

### 5.2.1 ISP命令格式

"命令 参数\_0 参数\_1 ... 参数\_n<CR><LF>" "数据" ("数据"只适用于写命令)。

### 5.2.2 ISP响应格式

"返回代码<CR><LF>响应\_0<CR><LF>响应\_1<CR><LF> ... 响应\_n<CR><LF>" "数据" ("数据"只适用于读命令)

### 5.2.3 ISP 数据格式

数据流采用 UU 编码格式。UU 编码算法将 3 字节二进制数据转换成 4 字节的可打印 ASCII 字符。该编码的效率高于 Hex 格式; Hex 格式将 1 字节二进制数据转换成 2 字节ASCII Hex数据。发送器在发送20个UU编码行之后发送校验和。任何UU编码行的长度都不应超过 61 个字符(字节), 也就是说它可以保持 45 个数据字节。接收器应当将该校验和与接收到的数据的校验和比较。如果校验和匹配, 接收器响应

“OK<CR><LF>”, 并等待下一次发送。如果校验和不匹配, 接收器则响应

“RESEND<CR><LF>”。作为响应, 发送器应当将字节重新发送。

UU-encode 的详细描述可以在 wotsit 网页得到。

### 5.2.4 ISP 流量控制

软件XON/XOFF流量控制方法用来防止由缓冲区溢出产生的数据丢失。当数据接收过快, 接收端将发送一个 ASCII控制符DC3(停止)停止数据流。发送 ASCII 控制符 DC1 (开始)则恢复数据流。主机也应支持相同的流量控制协议。

### 5.2.5 ISP 命令中止

通过发送 ASCII 控制字符 "ESC"可以终止命令。这个特殊字符没有被当作ISP命令写入 "ISP 命令s" 部分。一旦接收到 ESC 字符, ISP 命令处理程序将等待一个新的命令。

### 5.2.6 ISP期间的中断

系统复位之后, Flash引导区的中断向量表有效。

### 5.2.7 IAP期间的中断

在擦写操作期间, 片上flash 存储器不可访问。当用户Flash区激活后, 用户程序代码开始执行中断向量。在执行 flash 擦写 IAP 调用, 用户必须禁止中断, 或确保用户中断向量在 RAM中, 且中断处理程序也在RAM中。IAP代码不能使用和禁止中断

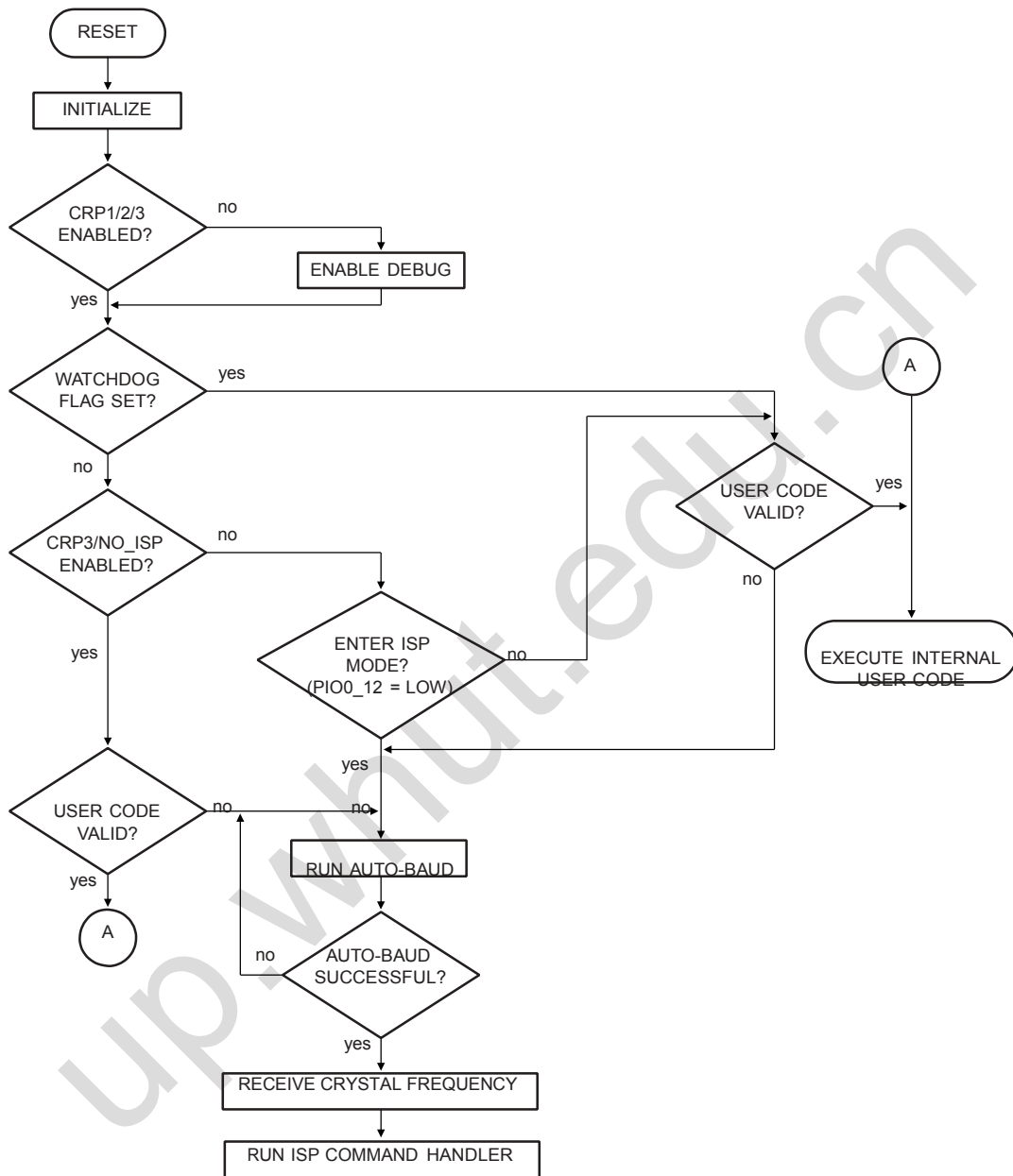
### 5.2.8 ISP 命令处理程序使用的 RAM

ISP 命令使用片上地址从 0x1000 017C 到 0x1000 025B 的RAM。用户可以使用该区域，但复位后内容可能会丢失。Flash 编程命令使用片上最顶端的32字节RAM。堆栈位于RAM 顶端 - 32。用户最大可使用的堆栈是256字节，堆栈是向下增加的。

### 5.2.9 IAP命令处理程序使用的 RAM

Flash编程命令使用片上RAM顶端的32字节。用户最大可使用的堆栈是128字节，堆栈是向下增加的。

### 5.3 启动过程流程图



(1) 需要更多设置晶振的细节，见 第239页 17-8.8 小节“Reinvoke ISP”

**图 50. 引导过程流图**

## 5.4 扇区编号

一些 IAP 和 ISP 命令在 "扇区" 操作并指定扇区编号。下表列出了 LPC111x 系列处理器中扇区编号和存储器地址之间的对应关系。

表 214. LPC111x 系列处理器扇区编号

扇区 编号	扇区 大小	存储器地址范围	LPC1111 8 kB flash	LPC1112 16 kB flash	LPC1113 24 kB flash	LPC1114 32 kB flash
0	4 kB	0x0000 0000 - 0x0000 0FFF	有	有	有	有
1	4 kB	0x0000 1000 - 0x0000 1FFF	有	有	有	有
2	4 kB	0x0000 2000 - 0x0000 2FFF	-	有	有	有
3	4 kB	0x0000 3000 - 0x0000 3FFF	-	有	有	有
4	4 kB	0x0000 4000 - 0x0000 4FFF	-	-	有	有
5	4 kB	0x0000 5000 - 0x0000 5FFF	-	-	有	有
6	4 kB	0x0000 6000 - 0x0000 6FFF	-	-	-	有
7	4 kB	0x0000 7000 - 0x0000 7FFF	-	-	-	有

## 6. 代码读保护 (CRP)

代码读保护是一种允许用户在系统上设置以不同的片上Flash安全级别访问以及限制ISP使用的保护机制。如果需要，程序可通过在 Flash 0x0000 02FC地址处写入一个特征值来调用CRP。代码读保护不影响IAP 命令。

注意：任何CRP 改动，必须是在系统复位后才会生效。

表 215. 代码读保护选项

名称	写入0x0000 02FC的模式值	描述
NO_ISP	0x4E69 7370	阻止 PIO0_1 引脚进入 ISP 模式。PIO0_1 可以正常行使其它功能.
CRP1	0x12345678	<p>禁止通过JTAG引脚访问芯片。该模式允许通过以下ISP命令更新部分flash，但受到限制：</p> <ul style="list-style-type: none"><li>• 写 RAM 命令不能访问地址 0x1000 0300 以下的 RAM 。</li><li>• 复制 RAM 到 flash 的命令不能写 0 扇区。</li><li>• 擦除命令，只有当所有扇区被擦除才可以擦除0扇区。</li><li>• 比较命令不可用。</li><li>• 读存储器命令不可用。</li></ul> <p>该模式在需要 CRP同时又需要flash区块更新、但不允许擦除所有扇区时非常有用。由于部分更新时比较命令不可用，次级引导区采用校验机制以确保flash的完整性。</p>
CRP2	0x87654321	<p>通过 JTAG 引脚访问芯片被禁止。以下 ISP命令不可用。</p> <ul style="list-style-type: none"><li>• 读存储器</li><li>• 写 RAM</li><li>• 执行</li><li>• 复制RAM 到 flash</li><li>• 比较</li></ul> <p>当CRP2被允许后，ISP擦除命令只能擦除用户扇区。</p>
CRP3	0x43218765	<p>禁止通过JTAG引脚访问芯片。如果扇区0位置包含有效的用户代码，通过拉低PIO0_1引脚来禁止进入ISP。</p> <p>该模式使用 PIO0_1引脚有效地禁止ISP重写。用户程序可使用IAP调用或重调用ISP命令提供flash更新机制，允许通过UART更新flash。</p> <p>注意: 如果写入 <b>CRP3</b>， 将不能在处理器上进行批量测试。</p>

表 216. 代码读保护 硬件/软件交互

CRP 选项	用户代码有效	PIO0_1 引脚复位时状态	允许JTAG	进入 ISP 模式	在 ISP模式下更新部分flash
无	否	x	是	是	是
无	是	高	是	否	NA
无	是	低	是	是	是
CRP1	是	高	否	否	NA
CRP1	是	低	否	是	是
CRP2	是	高	否	否	NA

表 216. 代码读保护 硬件/软件交互

CRP 选项	用户代码有效	PIO0_1 引脚 复位时状态	允许JTAG	进入 ISP 模式	在 ISP模式下 更新部分flash
CRP2	是	Low	否	是	否
CRP3	是	x	否	否	NA
CRP1	否	x	否	是	是
CRP2	否	x	否	是	否
CRP3	否	x	否	是	否

表 217. 不同的 CRP级别所允许的ISP命令

ISP 命令	CRP1	CRP2	CRP3 (不允许进入 ISP 模式)
解锁	允许	允许	n/a
设置波特率	允许	允许	n/a
回显	允许	允许	n/a
写RAM	允许; 仅限 0x1000 0300 之上	禁止	n/a
读存储器	禁止	禁止	n/a
准备写操作的扇区	允许	允许	n/a
复制 RAM 到 flash	允许; 禁止复制到扇区0	禁止	n/a
执行	禁止	禁止	n/a
擦除扇区	允许; 仅当所有扇区被删除 时，可以删除扇区0。	允许; 仅限所有 扇区。	n/a
空扇区检查	禁止	禁止	n/a
读分类ID	允许	允许	n/a
读Boot代码版本	允许	允许	n/a
比较	禁止	禁止	n/a
读 UID	允许	允许	n/a

在选定CRP模式后，允许通过ISP访问芯片，不支持和受限制的ISP 命令将被终止，并返回代码 CODE\_READ\_PROTECTION\_ENABLED。

6.1 ISP 进入保护

除了上述三种CRP模式，用户可以阻止通过PIO0\_1引脚进入ISP模式，从而释放PIO0\_1引脚用于其他功能。这通常叫做NO\_ISP模式。通过向地址 0x0000 02FC 写入特征值0x4E69 7370 进入NO\_ISP模式。



7. ISP 命令

以下命令是 ISP 命令处理程序所接受的命令。每个命令都有具体的状态码。当接收到未定义的命令时，命令处理程序返回代码INVALID\_COMMAND。命令和返回代码为ASCII格式。

只有当接收到的 ISP 命令执行完毕时，ISP 命令处理程序才发送CMD\_SUCCESS。这时主机才能发送新的 ISP 命令。“设置波特率”、“写 RAM”、“读存储器”和“运行”命令除外。

表 218. ISP 命令总览

ISP 命令	说法	描述
解锁	U <解锁码>	<a href="#">表 17-219</a>
设置波特率	B <波特率> <停止位>	<a href="#">表 17-220</a>
回显	A <设定>	<a href="#">表 17-221</a>
写 RAM	W <起始地址> <字节数>	<a href="#">表 17-222</a>
读存储器	R <地址> <字节数>	<a href="#">表 17-223</a>
准备写操作的扇区	P <起始扇区编号> <结束扇区编号>	<a href="#">表 17-224</a>
复制 RAM 到 flash	C <Flash 地址> <RAM 地址> <字节数>	<a href="#">表 17-225</a>
执行	G <地址> <模式>	<a href="#">表 17-226</a>
擦除扇区	E <起始扇区编号> <结束扇区编号>	<a href="#">表 17-227</a>
空扇区检查	I <起始扇区编号> <结束扇区编号>	<a href="#">表 17-228</a>
读分类ID	J	<a href="#">表 17-229</a>
读Boot代码版本	K	<a href="#">表 17-231</a>
比较	M <地址1> <地址2> <字节数>	<a href="#">表 17-232</a>
读UID	N	<a href="#">表 17-233</a>

7.1 解锁 <解锁码>

表 219. ISP解锁命令

命令	U
输入	解锁码: 23130 <sub>10</sub>
返回码	CMD_SUCCESS   INVALID_CODE   PARAM_ERROR
描述	该命令用于解锁Flash写、擦除和运行命令。
举例	"U 23130<CR><LF>" 解锁Flash读写、擦除&运行命令。

7.2 设置波特率 <波特率> <停止位>

表 220. ISP 设置波特率命令

命令	B
输入	波特率: 9600   19200   38400   57600   115200   230400 停止位: 1   2
返回码	CMD_SUCCESS   INVALID_BAUD_RATE   INVALID_STOP_BIT   PARAM_ERROR
描述	该命令用于改变波特率。新的波特率在命令处理程序发送返回代码 CMD_SUCCESS之后生效。
举例	"B 57600 1<CR><LF>" 设置串口波特率57600bps和1个停止位。

7.3 回显 <设定>

表 221. ISP回显命令

命令	A
输入	Setting: ON = 1   OFF = 0
返回码	CMD_SUCCESS   PARAM_ERROR
描述	回显命令的默认设定是ON。当ON时，ISP命令处理程序将接收到的串行数据发送回主机。
举例	"A 0<CR><LF>" 关闭回显。

7.4 写RAM <起始地址> <字节数>

主机应当在接收到CMD\_SUCCESS返回代码后发送数据。主机应当在发送20个UU编码行之后发送校验和。校验和通过累加原始数据（UU编码之前）获得，在传输完20行UU 编码后被重置。任何UU编码行的长度都不应超过 61个字符（字节），即它可以保持45 个数据字节。当数据少于 20 个 UU编码行时，按照实际发送的字节数计算校验和。ISP命令处理器程序会将它与接收字节的校验和相比较。若校验和匹配，那么主机响“OK<CR><LF>”，并等待下一次发送。如果校验和不匹配，主机响应“RESEND<CR><LF>”，作为响应ISP命令处理程序将重新发送字节。

表 222. ISP 写 RAM 命令

命令	W
输入	<b>起始地址:</b> 被写RAM的起始地址，该地址边界应当字对齐。 <b>字节数:</b> 写入的字节数，字节数应当为4的倍数。
返回码	CMD_SUCCESS   ADDR_ERROR (地址边界不是字对齐)   ADDR_NOT_MAPPED   COUNT_ERROR (字节数不是 4 的整数倍)   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED
描述	该命令用于将数据下载到RAM。数据应当为UU编码格式。该命令在启用代码读保护时被禁止。
举例	"W 268436224 4<CR><LF>" 写 4 字节数据到地址 0x1000 0300 处。

7.5 读存储器 <地址> <字节数>

数据流紧随命令成功返回码之后。发送完 20个UU编码行之后发送校验和。校验和是通过累加原始数据（UU编码之前）产生的，在传输完 20行 UU 编码后被重置。任何UU编码行的长度都不超过61个字符（字节），即它可以保持45 个数据字节。当数据少于20个UU编码行时，按照实际发送的字节数计算校验和。主机将它与接收字节的校验和相比较。如果校验和匹配，那么主机响“OK<CR><LF>”，并等待下一次发送。如果校验和不匹配，主机响应“RESEND<CR><LF>”。作为响应，ISP命令处理程序将重新发送字节。

表 223. ISP 读存储器命令

命令	R
输入	<b>起始地址:</b> 被读出数据字节的地址，该地址边界应当字对齐。 <b>字节数:</b> 读出的字节数。计数值应当为4的倍数。
返回码	CMD_SUCCESS 后面跟 <实际数据 (UU-编码)>   ADDR_ERROR (地址边界不是字对齐的)   ADDR_NOT_MAPPED   COUNT_ERROR (字节数不是4的整数倍)   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED
描述	该命令用于读出RAM或Flash存储器的数据。该命令在启用代码读保护时被禁止。
举例	"R 268435456 4<CR><LF>" 从地址 0x1000 0000处读 4 字节 数据。

7.6 准备写操作作用的扇区 <起始扇区号> <结束扇区号>

该命令将Flash写/擦除操作分成两个步骤处理。

表 224. ISP 准备写操作作用的扇区命令

命令	P
输入	起始扇区号 结束扇区号: 应当大于等于起始扇区号
返回码	CMD_SUCCESS   BUSY   INVALID_SECTOR   PARAM_ERROR
描述	该命令必须在执行“复制RAM到Flash”或“擦除扇区”命令之前执行。这两个命令的成功执行会导致相关的扇区再次被保护。该命令不能用于boot扇区。如果要准备单个扇区，可将起始和结束扇区号设置为相同值。
举例	"P 0 0<CR><LF>" 准备Flash扇区0。

7.7 复制 RAM 到 flash <Flash 地址> <RAM 地址> <字节数>

表 225. ISP复制命令

命令	C
输入	Flash 地址(DST): 要写入数据字节的目标Flash地址。目标地址的边界应当为256字节对齐。 RAM 地址(SRC): 读出数据字节的源RAM地址。 字节数: 写入字节的数目。应当为 256   512   1024   4096
返回码	CMD_SUCCESS   SRC_ADDR_ERROR (地址边界不是字对齐)   DST_ADDR_ERROR (地址边界错误)   SRC_ADDR_NOT_MAPPED   DST_ADDR_NOT_MAPPED   COUNT_ERROR (字节计数值不是256   512   1024   4096)   SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION   BUSY   CMD_LOCKED   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED
描述	该命令用于对Flash存储器编程。受影响的扇区应当先通过调用“准备写操作作用的扇区”命令来做准备。当成功执行复制命令后，扇区将自动受到保护。该命令不能写boot扇区。该命令在启用代码读保护时被禁止。
举例	"C 0 268467504 512<CR><LF>" 从 RAM 地址 0x1000 0800 复制 512 字节数据到flash 地址 0。

7.8 执行 <地址> <模式>

表 226. ISP 执行命令

命令	G
输入	地址：代码执行的起始Flash或RAM地址。该地址边界应当以字对齐。 模式：： T（执行Thumb模式下的程序） A（执行ARM模式下的程序）。
返回码	CMD_SUCCESS   ADDR_ERROR   ADDR_NOT_MAPPED   CMD_LOCKED   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED
描述	该命令用于执行（调用）位于 RAM 或 Flash 存储器中的程序。一旦成功执行该命令，就有可能不再返回ISP命令处理程序。该命令在启用代码读保护时被禁止。
举例	"G 0 A<CR><LF>" 跳转到ARM模式下的地址 0x0000 0000 处。

7.9 擦除扇区 <起始扇区编号> <结束扇区编号>

表 227. ISP擦除扇区命令

命令	E
输入	起始扇区编号 结束扇区编号：应当大于等于起始扇区号。
返回码	CMD_SUCCESS   BUSY   INVALID_SECTOR   SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION   CMD_LOCKED   PARAM_ERROR   CODE_READ_PROTECTION_ENABLED
描述	该命令用于擦除片内 Flash 存储器的一个或多个扇区。boot扇区不能由该命令擦除。该命令在启用代码读保护时只允许擦除所有用户扇区。
举例	"E 2 3<CR><LF>" 擦除 flash 扇区 2 和 3。

7.10 空扇区检查<起始扇区编号> <结束扇区编号>

表 228. ISP 空扇区检查命令

命令	I
输入	起始扇区编号 结束扇区编号： 应当大于等于起始扇区号。
返回码	CMD_SUCCESS   SECTOR_NOT_BLANK (后面跟<第一个非空字的偏移量> <非空字的内容>)  INVALID_SECTOR   PARAM_ERROR
描述	该命令用于对片内Flash存储器的一个或多个扇区进行查空。 由于扇区 0的前 64 字节重新映射到 Flash boot 扇区，因此对其进行查空一定会失败。
举例	"I 2 3<CR><LF>" 对Flash扇区2和 3进行查空。

7.11 读分类ID号

表 229. ISP读分类ID号

命令	J
输入	无
返回码	CMD_SUCCESS后跟ASCII格式的分类ID号。(见 表 17-230 “LPC111x 分类ID号”).
描述	该命令用于读取器件的分类ID号。

表 230. LPC111x系列处理器分类ID号

处理器	ASCII/dec 编码	Hex 编码
LPC1111FHN33/101	69095467	0x041E 502B
LPC1111FHN33/201	68571179	0x0416 502B
LPC1112FHN33/101	70078507	0x042D 502B
LPC1112FHN33/201	69554219	0x0425 502B
LPC1113FHN33/201	70537259	0x0434 502B
LPC1113FHN33/301	70520875	0x0434 102B
LPC1113FBD48/301	70520875	0x0434 102B
LPC1114FHN33/201	71585835	0x0444 502B
LPC1114FHN33/301	71569451	0x0444 102B
LPC1114FBD48/301	71569451	0x0444 102B
LPC1114FA44/301	71569451	0x0444 102B

7.12 读Boot 代码版本号

表 231. ISP 读 Boot 代码版本号命令

命令	K
输入	无
返回码	CMD_SUCCESS 后跟2字节ASCII格式的boot代码版本号 将其解释为<字节1 (主)>.<字节0 (次)>。
描述	该命令用于读取boot代码版本号。

7.13 比较 <地址1> <地址2> <字节数>

表 232. ISP 比较命令

命令	M
输入	<b>地址1 (DST):</b> 要比较的目标Flash或RAM的起始地址。该地址边界应当字对齐。 <b>地址2 (SRC):</b> 要比较的源Flash或RAM的起始地址。该地址边界应当字对齐。 <b>字节数:</b> 待比较的字节数，计数值应当为4的倍数。
返回码	CMD_SUCCESS   (源和目标的数据相同) COMPARE_ERROR   (后面跟第一个不匹配字节的地址) COUNT_ERROR (字节数不是4的倍数)   ADDR_ERROR   ADDR_NOT_MAPPED   PARAM_ERROR
描述	该命令用来比较两段的存储器中的内容。 当源或目标地址是从地址 0开始的前512字节中的任意一个时，比较结果可能不正确。前512个字节重新映射到boot ROM。
举例	"M 8192 268468224 4<CR><LF>"将0x108000开始的RAM中的4个字节与0x2000开始的Flash中的4个字节相比较。

7.14 读 UID

表 233. 读 UID 命令

命令	N
输入	无
返回码	CMD_SUCCESS 后跟4个32位ASCII码的晶元测试信息。字的低地址部分先发送。
描述	该命令用来读取UID。

7.15
 ISP返回码

表 234.
 ISP返回码总览

返回码	助记符	描述
0	CMD_SUCCESS	命令被成功执行。只有当主机发出的命令被成功执行完毕后，才由 ISP 处理程序发出。
1	INVALID_COMMAND	无效命令。
2	SRC_ADDR_ERROR	源地址边界没有字对齐。
3	DST_ADDR_ERROR	目标地址边界没有字对齐。
4	SRC_ADDR_NOT_MAPPED	源地址没有位于存储器映射中。计数值必须考虑是否可用。
5	DST_ADDR_NOT_MAPPED	目标地址没有位于存储器映射中。计数值必须考虑是否可用。
6	COUNT_ERROR	字节计数值不是 4 的倍数或是一个非法值。
7	INVALID_SECTOR	扇区号无效或结束扇区号小于起始扇区号。
8	SECTOR_NOT_BLANK	扇区非空。
9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	为写操作准备扇区命令未执行。
10	COMPARE_ERROR	源和目标数据不相等。
11	BUSY	Flash编程硬件接口忙。
12	PARAM_ERROR	参数不足或无效参数。
13	ADDR_ERROR	地址边界没有字对齐。
14	ADDR_NOT_MAPPED	地址没有位于存储器映射中。计数值必须考虑是否可用。
15	CMD_LOCKED	命令被锁定。
16	INVALID_CODE	解锁码无效。
17	INVALID_BAUD_RATE	无效的波特率设定。
18	INVALID_STOP_BIT	无效停止位设定。
19	CODE_READ_PROTECTION_ENABLED	允许代码读保护。

8.
 IAP命令

对于IAP来说，应当通过寄存器 r0 中的字指针指向存储器(RAM)包含的命令代码和参数来调用 IAP 程序。IAP 命令的结果返回到寄存器 r1 所指向的结果表。用户可以通过传送相同的指针到寄存器 r0 和 r1 重用命令表得到结果。参数表应当足够大，以保证在结果数多余参数数时仍能容纳所有的结果。参数传递见图17-51。命令参数和结果的数目根据 IAP命令而有所不同。命令参数的最大数目为5个，是“复制RAM到Flash”命令。



命令结果的最大数目为4, 是“读UID”命令的返回。命令处理程序在接收到一个未定义的命令时, 会发送状态码NVALID\_COMMAND。IAP命令是 thumb代码, 通常位于地址 0x1FFF 1FF0 处。

IAP功能可用下面的C代码来调用。

定义IAP程序的入口地址。由于IAP地址的第 0 位是 1, 因此当程序计数器转移到该地址时会转换到Thumb指令集。

```
#define IAP_LOCATION 0x1fff1ff1
```

定义数据结构或指针, 将 IAP命令表和结果表传递给 IAP函数:

```
unsigned long command[5];  
unsigned long result[4];
```

或

```
unsigned long * command;  
unsigned long * result;  
command=(unsigned long *) 0x.....  
result= (unsigned long *) 0x.....
```

定义函数类型指针, 函数包含2个参数, 无返回值。注意: IAP将结果表的基地址放于 R1中, 以返回结果。

```
typedef void (*IAP)(unsigned int [], unsigned int []);  
IAP iap_entry;
```

设置函数指针:

```
iap_entry=(IAP) IAP_LOCATION;
```

需要调用 IAP时, 使用下面的语句。

```
iap_entry (command, result);
```

一些IAP调用需要4个以上参数。按照ARM 规范 (The ARM Thumb Procedure Call Standard SWS ESPC0002 A-05)规定, 最多有4个参数可以通过通用寄存器r0、r1、r2和r3传递, 剩下的参数通过堆栈传递。最多有4个参数可以通过r0、r1、r2和r3 寄存器返回, 其他参数直接通过存储器返回。如果使用ARM建议的参数传递和返回机制, 不同供应商提供的不同C编译器可能产生问题。采用上面建议的参数传递机制可以减少那样的风险。

Flash存储器在擦写操作期间不可访问。执行flash 写/擦除操作的 IAP 命令使用片内 RAM 顶端的 32 个字节空间; 如果应用程序中允许 IAP 编程, 那么用户程序不应使用这段空间。

表 235. IAP 命令总览

IAP 命令	命令码	描述
准备用于写操作的扇区	50 <sub>10</sub>	<a href="#">表 17-236</a>
复制 RAM 到 flash	51 <sub>10</sub>	<a href="#">表 17-237</a>
擦除扇区	52 <sub>10</sub>	<a href="#">表 17-238</a>
检查空扇区	53 <sub>10</sub>	<a href="#">表 17-239</a>
读分类ID	54 <sub>10</sub>	<a href="#">表 17-240</a>
读 Boot 代码版本	55 <sub>10</sub>	<a href="#">表 17-241</a>
比较	56 <sub>10</sub>	<a href="#">表 17-242</a>
重调用 ISP	57 <sub>10</sub>	<a href="#">表 17-243</a>
读 UID	58 <sub>10</sub>	<a href="#">表 17-244</a>

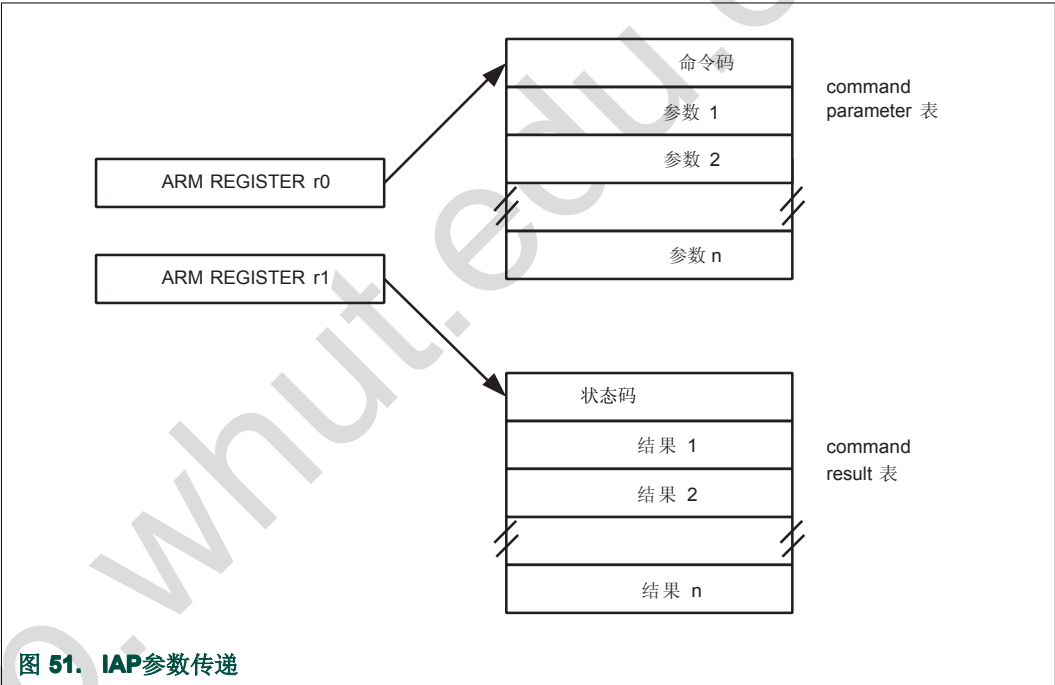


图 51. IAP参数传递

8.1 准备用于写操作的扇区

该命令将Flash写/擦除操作分两步执行。

表 236. IAP 准备用于写操作的扇区命令

命令	准备用于写操作的扇区
输入	命令码: 50 <sub>10</sub> 参数0: 起始扇区编号 Param1:结束扇区编号 (应当大于等于起始扇区编号)。

表 236. IAP 准备用于写操作的扇区命令

命令	准备用于写操作的扇区
返回码	CMD_SUCCESS   BUSY   INVALID_SECTOR
结果	无
描述	该命令必须在执行“复制RAM到Flash”或“擦除扇区”命令之前执行。这两个命令的成功执行会导致相关的扇区再次被保护。该命令不能用于boot扇区。如果要准备单个扇区，可将起始和结束扇区号设置为相同值。

8.2 复制 RAM 到 flash

表 237. IAP Copy RAM to flash command

命令	复制 RAM 到 flash
输入	命令码: 51 <sub>10</sub>  参数0(DST): : 要写入数据字节的目标Flash的地址。目标地址的边界应为256字节对齐。  参数1(SRC): 读出数据字节的源RAM的地址。该地址边界应当字对齐。  参数2: 写入字节的数目。应当是 256   512   1024   4096。  参数3: 系统时钟频率 (CCLK) 单位: kHz。
返回码	CMD_SUCCESS   SRC_ADDR_ERROR (地址边界不是字对齐)   DST_ADDR_ERROR (地址边界错误)   SRC_ADDR_NOT_MAPPED   DST_ADDR_NOT_MAPPED   COUNT_ERROR (字节数不是 256   512   1024   4096)   SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION   BUSY
结果	无
描述	该命令用于对Flash存储器编程。被编程的扇区应当先通过调用“准备写操作的扇区”命令进行准备。当成功执行复制命令后，扇区将自动受到保护。该命令不能写boot扇区。

8.3 擦除扇区

表 238. IAP 擦除扇区命令

命令	擦除扇区
输入	命令码: <b>52<sub>10</sub></b> 参数0: 起始扇区编号 参数1: 结束扇区编号 (应当大于等于起始扇区编号)。 参数2: 系统时钟频率 (CCLK)，单位: kHz。
返回码	CMD_SUCCESS   BUSY   SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION   INVALID_SECTOR
结果	无
描述	该命令用于擦除片内Flash存储器的一个或多个扇区。 <b>boot</b> 扇区不能由该命令擦除。要擦除单个扇区可将起始和结束扇区号设定为相同值。

8.4 检查空扇区

表 239. IAP 检查空扇区命令

命令	检查空扇区
输入	命令码: <b>53<sub>10</sub></b> 参数0: 起始扇区编号。 参数1: 结束扇区编号(应当大于起始扇区编号)。
返回码	CMD_SUCCESS   BUSY   SECTOR_NOT_BLANK   INVALID_SECTOR
结果	<b>Result0:</b> 若状态代码为SECTOR_NOT_BLANK，则是第一个非空字位置的偏移量。 <b>Result1:</b> 非空字位置的内容。
描述	该命令用于对片内Flash存储器的一个或多个扇区进行查空。要对单个扇区查空，可将起始和结束扇区号设定为相同值。

8.5 读分类ID

表 240. IAP 读分类ID

命令	读分类ID号
输入	命令码: <b>54<sub>10</sub></b> 参数: 无
返回码	CMD_SUCCESS
结果	<b>结果0:</b> 分类ID。
描述	该命令用于读分类ID。

8.6 读 Boot 代码版本号

表 241. IAP 读 Boot 代码版本号命令

命令	读 boot 代码版本号
输入	命令码: <b>55<sub>10</sub></b> 参数: 无
返回码	CMD_SUCCESS
结果	<b>结果0:</b> 2字节boot代码版本号。将其解释为<字节1 (主)>.<字节0 (次)>。
描述	该命令用于读取boot代码版本号。

8.7 比较 <地址1> <地址2> <字节数>

表 242. IAP比较命令

命令	比较
输入	命令码: <b>56<sub>10</sub></b> 参数 <b>0(DST)</b> : 要比较的Flash或RAM的数据字节的起始地址。该地址边界应当字对齐。 参数 <b>1(SRC)</b> : 要比较的Flash或RAM的数据字节的源起始地址。该地址边界应当字对齐。 参数 <b>2</b> : 待比较的字节数。该值应当为4的倍数。
返回码	CMD_SUCCESS   COMPARE_ERROR   COUNT_ERROR (字节数不是 4 的整数倍)   ADDR_ERROR   ADDR_NOT_MAPPED
结果	<b>结果0:</b> 当状态代码为COMPARE_ERROR时，为第一个不匹配字节的偏移地址。
描述	该命令用来比较两块存储器中的内容。 当源或目标地址包含从地址 <b>0</b> 开始的前 <b>512</b> 字节中的任意一个时，比较的结果不一定正确。前 <b>512</b> 字节可以被重新映射到 <b>RAM</b> 中。

8.8 重调用 ISP

表 243. 重调用 ISP

命令	重调用ISP
输入	命令码: <b>57<sub>10</sub></b>
返回码	无。
结果	无。
描述	该命令用于调用引导程序进入ISP 模式。它会映射引导向量，设置 PCLK = CCLK，配置 UART RXD 和 TXD 引脚，复位计数器/定时器 CT32B1，复位 U0FDR (见 表 9-132)寄存器。当有效用户程序在内部 Flash中，且无法通过 PIO0_1引脚无法访问时，可使用该命令进入ISP 模式。

8.9 读UID

表 244. IAP 读 UID 命令

命令	读 UID
输入	命令码: 5810
返回码	CMD_SUCCESS
结果	结果0: 第一个 32-位字 (在低地址)。 结果1: 第二个 32-位字。 结果2: 第三个 32-位字。 结果3: 第四个 32-位字。
描述	该命令用来读 UID。

8.10 IAP状态码

表 245. IAP状态代码总览

状态码	助记符	描述
0	CMD_SUCCESS	命令成功执行。
1	INVALID_COMMAND	无效命令。
2	SRC_ADDR_ERROR	源地址边界没有字对齐。
3	DST_ADDR_ERROR	目标地址的边界错误。
4	SRC_ADDR_NOT_MAPPED	源地址没有位于存储器映射中。 计数值必须考虑是否可用。
5	DST_ADDR_NOT_MAPPED	目标地址没有位于存储器映射中。计数值必须考虑是否可用。
6	COUNT_ERROR	字节计数值不是4的倍数或是一个非法值。
7	INVALID_SECTOR	扇区号无效。
8	SECTOR_NOT_BLANK	扇区非空。
9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	“准备用于写的扇区”操作命令没有执行
10	COMPARE_ERROR	目的和源数据不相同。
11	BUSY	Flash 编程硬件接口忙。

9. 调试说明

9.1 比较flash映像

根据使用的调试器和当前IDE的调试设置，当调试器连接时可见的存储器可能是引导ROM、内部SRAM或flash。为了帮助确定当前调试环境下是哪块存储器，可以查看Flash地址 0x0000 0004 处的值。该地址包含进入ARM Cortex-M0 向量表代码的入口点，可以是引导ROM、内部SRAM或flash存储器的底端。

表 246. 调试模式的存储器映射

存储器映射模式	地址 0x0000 0004 处的起始地址
引导加载模式	0x1FFF 0000
用户 flash 模式	0x0000 0000
用户 SRAM 模式	0x1000 0000

9.2 串行线调试 (SWD) flash 编程接口

调试工具能够将部分Flash 内容写入 RAM，然后从适当的偏移地址重复执行执行IAP调用"复制 RAM 到 flash"。

10. Flash存储器访问

根据系统时钟频率，可以通过设置地址为0x4003 C010的FLASHCFG 寄存器来配置不同的 Flash 访问时间。

注意：不恰当的寄存器设置可能导致LPC111x不正确的Flash 存储器操作。

表 247. 配置寄存器(FLASHCFG, 地址 0x4003 C010)位域描述

位	符号	值	描述	复位值
1:0	FLASHTIM		Flash 存储器访问时间。 FLASHTIM +1 等于访问Flash 的系统时钟数。	10
		00	1 个系统时钟的lash 访问时间 (针对最高为20 MHz 系统频率)。.	
		01	2 个系统时钟的flash 访问时间(针对最高为40 MHz 系统频率)。.	
		10	3个系统时钟的flash 访问时间(针对最高为 50 MHz 系统频率)。.	
		11	保留。	<tbd>
31:2	-	-	保留。用户程序不应该更改这些位。这些位必须以读出的原值写回。	